User's guide of g-expectation and reflected backward stochastic differential equation

Mingyu XU [a,b*]

[a]School of Mathematics and System Science, Shandong University,

250100, Jinan, China

[b]Institute of Applied Mathematics, AMSS, CAS, 100190, Beijing, China

1 Backward Stochastic Differential Equation: g-Expectation

Let $(\Omega, \mathcal{F}, P)$ be a probability space, and $\{(B_s)_{s\geq 0}\}$ be a 1-dimensional Brownian motion defined on this space. Set $\{\mathcal{F}_t; 0 \leq t \leq T\}$ be the naturel filtration generated by the Brownian motion (B_t) :

$$\mathcal{F}_t = \sigma\{B_s; 0 \leq s \leq t\},$$

where $\mathcal{F}_0$ contained all $P-$null sets of $\mathcal{F}$. All processes metioned in this paper are supposed to be $\mathcal{F}_t$-adapted, and we are interested in the behavior of processes on a given interval $[0, T]$. The space $L^2(0, T; R^d)$ is the space of $\mathcal{F}_t$-adapted square-integrable process on the interval $[0, T]$, the space $H^2(0, T; R^d)$ is the space of $\mathcal{F}_t$-predictable square-integrable process on the interval $[0, T]$, and L_T^2 is the space of $\mathcal{F}_T$-measurable square-integrable random variable.

The linear backward stochastic differential equations (BSDE in short) is firstly introduced by J.M.Bismut in 1973 [2]and systematically studied by A. Bensousan in order to obtain the stochastic maximum principle of Pontryagin's type of optimal control systems. E.Pardoux and S. Peng, in 1990 firstly proved the uniqueness and existence of the solution to backward stochastic

*Email: xumy@amss.ac.cn

differential equation[5], that is, there exists a unique pair of $\mathcal{F}_t$-processes $(Y_t, Z_t) \in L^2(0,T;R) \times H^2(0,T;R^d)$, satisfied the following equation

$$Y_t = \xi + \int_t^T g(t, Y_r, Z_r)dr - \int_t^T Z_r dB_r.$$

where g satisfied (i) $g : [0,T] \times R \times R \to R$, and $g(\cdot, 0, 0) \in L^2$,(ii) the Lipschitz condition: $\forall (y_1, z_1), (y_2, z_2) \in R \times R^d$, there exists a constant $C > 0$, satisfying

$$|g(t, y_1, z_1) - g(t, y_2, z_2)| \leq C(|y_1 - y_2| + |z_1 - z_2|),$$

and $\xi \in L_T^2$.

Although the history of the theory of backward stochastic differential equation is short, but it is developing very fast. For itself has many interesting properties as well as many useful applications. It can solve problems in partial differential equation theory, differential geometry, and mathematics finance. Using backward stochastic differential equations, Professor Peng gave a definition: g-expectation, which extend the definition of classical expectation. And it can be used widely in Mathematical finance.

For linear BSDEs, we can solve them explicitly by using duality method. But for nonlinear case, unfortunately, the equations have no explicit solution. Thus the method of numerical solutions are in order. A numerical algorithms had been proposed and calculated in 1994 by the research group of BSDE, under the direction of S.Peng, in Shangdong University. And we obtain numerical results of BSDEs, and the convergence of this method has been proved by Philippe Briand, Bernard Delyon and Jean Mémin(2001) in [1].

Using this method, we have developed a user-interface of programs for calculations and simulations of BSDEs. With this user-interface people who have no experience in stochastic calculations and numerical simulations could quickly learn how to use our programs to calculate, to simulate and to study the BSDEs. Briefly speaking, after inputting his parameters, i.e., the function $g = g(y, z)$, the terminal condition ξ in our BSDE, one can type them into our user-interface then execute and simulate the BSDE by clicking the related buttons on the user-interface shown in fig.1.1. We think that the visibility of the numerical results of our programs will be useful for specialists and non specialists in BSDE. Such knid of user-interface is firstly developped by Zhou Haibin from 1996. These program that we put on the website are made by Xu Mingyu from 2000.

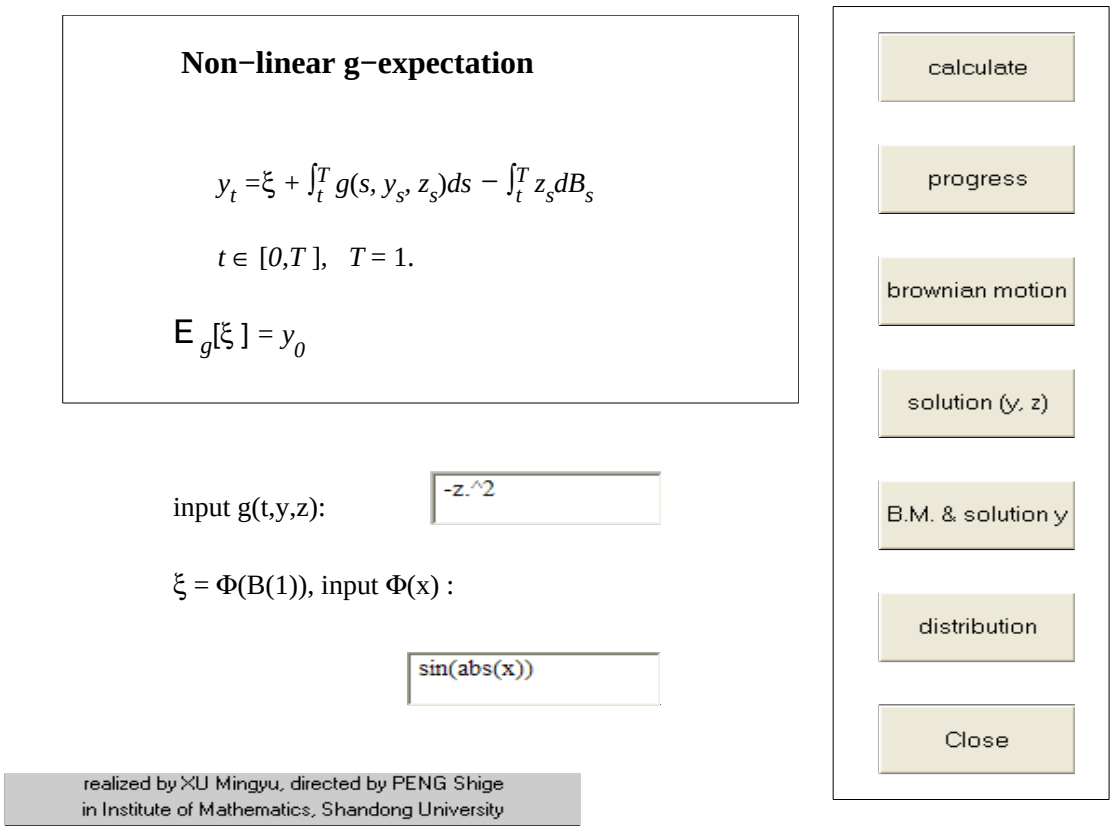

Figure 1:Interface for g-expectation(BSDE)

In fact this group of programs can be used to calculate the BSDE, not only the g-expectation, since the g-expectation is a specialized BSDE with a condition of g s.t. $g(y, 0) \equiv 0$ for $\forall y \in R$.

1.1 User's guide for BSDE: g-expectations

The programs are realized by Matlab's *.p files and is compressed as Xugeshow.zip. To run these programs, Matlab 5.3 or higher version is required. After download the compressed file, you should:

1. uncompress the file Xugeshow.zip in the document C: \matlab \work (or D:\..., your Matlab is installed in the hard disk D:\).

2. Run the Matlab command window.

3. Then within this window single click "File" in the menu buttons and then, among the prompted file buttons, single click "Set Path" button. Then

within the prompted "Path Browse" window, browse and add the direction C:\matlab\work\gexp in the Path.

4. After these preparation, you can run our program: in Matlab's command window, type "gexpectation" followed with a "return". Then the g–expectation window (figure No.1) is prompted.

An important feature of this program is its strong capacity of user-interface. Run command in the command window. We get the main user-interface shown in fig.1.

At the left side of the user-interface, the g-expectation is shown on the upside to indicate you the meaning of input functions. On the downside, you can input two functions: the generator g and the terminal condition ξ. For example, in fig.1.1 we will input: $g(t, y, z) = -z^2$, $\xi = \Phi(B_1) = \sin(|B_1|)$. So in the blank spaces we type g=-z.^2, Φ=sin(abs(x)). Here we offer a table of the transform between the formulas and the expressions in Matlab, the left are the fomula in mathematics, the right are the expression in Matlab.

Multiply ($\times$)	.*
Divide ($\div$)	./
The power of n (x^n)	x.^n
Absolute value ($\|x\|$)	abs(x)
Squrare root ($\sqrt{x}$)	sqrt(x)
Exponential (e^x)	exp(x)
Natural logarithm ($\log x$)	log(x)
Sine ($\sin(x)$)	sin(x)

The plus and minus are just + and -, respectively. And for the trigonometric functions, the expressions are also like usual, so we did not list them out in the table.

For the coefficient g, the function only depend on t, y, z. It can not support other variable expect t, y, z. And for the terminal condition Φ, it's same, Φ only can have one variable x.

After inputting the parameters, you can use these programs to do the calculation.

1. Clicking the button 'calculate' on the left-side once, then the program of calculation will run. For getting the result, calculation will take certain seconds and then indicate the end of the calculation by jumping a dialogue-box "the calculation is complete." .

2. Click the next button "progress", the program will generate a new figure, named "calculating process of solution y". On this figure, you will see

the backward computation procedure of the function $y(t,x)$, dynamically and backwardly. In the figure the red lines above (resp. blue lines below) show the solution y (resp. Brownian motion). At the end, we can see a vertical red line, which simify the value of solution y at time 0. Then the colorful surface of solution y is generated in a new figure, which is named "surface for solution y". The green lines in this figure show the relation between the solution y and the Brownian motion, while the blue line below indicate the range of descret Brownian motion. By clicking the button "right", "up", you can see the surface in different direction.

The next three buttons on the main user-interface are for the simulations of the solved (y_t, z_t) .

3. Click the button "Brownian motion", a dynamically generated Brownian path will appear on the new figure, 'Sample way of Brownian Motion'. This path will terminated by a jump of a vertical line indicating the terminal value $y_T = \xi(\omega)$ of this sample. If you click the button "more" on this new figure, then another Brownian path and the related $\xi(\omega)$ will be produced in a different color.

4. Click the button "solution (y,z)" , you will see a moving (y_t, z_t) on the generated new figure "trajetories of solution y ". In the 1st (resp. the 2nd) column we show the trajetory of (B_t, y_t) (resp. (B_t, z_t)). On the above they are showed by a 3-d moving image, the red (resp. blue) lines show a trajetory of the solution y (resp. Brownian motion), and the light red vertical lines indicate the relation between the two trajetories. On the below, this trajetory of solution y is showed in 2-d moving image by a red line, with time being the x-coordinate. Clicking "more" button on this figure, there will produce a different couple (y_t, z_t) corresponding to a different Brownian motion path. Like in the figure for the solution surface the button "center", "right" and "up" are for see the two 3-d image in different direction.

5. Clicking the following button "B.M. and solution y", you will get a new figure, "solution y on the surface", for the sample way of solution y on the solution surface. On the above window of it, a trajectory of Brownian Motion $B_t(\omega)$ is showed on the ground, while the solution y_t according to this Brownian motion is showed on the solution surface. The button "more" is still for a new group of lines produced in the same color, and the button "right", "up" are for the same use like the ones in the figure "surface for solution y".

6. Click the next button "distribution", a new figure "distribution function" is generated to show the distribution function of solution y at different

time t. On this figure, from the time $T = 1$ to $t = 0$, the distribution functions of solution y are showed by lines in different color by turn. The button "right", "up" are for the same use.

7. For closing the figures there is two ways. One is using the button "Close" on them. The other is to click the little cross on the right-up corner of the figure.

In all the images,the meanings of the coordinate axises are noted directly beside the corresponding axises. So we omit the detailed explanation.

2 Reflected Backward Stochastic Differential Equations

2.1 Introduction for reflected backward differential equations

The reflected backward stochastic differential equations (RBSDE in short) are firstly introduced by N. El Karoui, C. Kapodjian, E. Pardoux, S. Peng and M.C. Quenez in [3]. We need the following notation,

$$S^2[0,T] = \left\{ \{\varphi_t, 0 \le t \le T\} \text{ is a 1-d } \mathcal{F}_t\text{-progressively process} s.t. E(\sup_{0\le t\le T} |\varphi_t|^2) < +\infty \right\}.$$

$$S_i^2[0,T] = \left\{ \{\varphi_t, 0 \le t \le T\} \text{ is a 1-d continuous increasing process } s.t. E(|\varphi_T|^2) < +\infty \right\}$$

With assumptions: (i) a terminal condition $\xi \in L_T^2$; (ii) a map g, which satisfied $g : [0,T] \times R \times R \to R$, $g(\cdot, 0, 0) \in L^2$,and the Lipschitz condition: $\forall (y_1, z_1), (y_2, z_2) \in R \times R^d$, there exists a constant $C > 0$, satisfying

$$|g(t, y_1, z_1) - g(t, y_2, z_2)| \le C(|y_1 - y_2| + |z_1 - z_2|);$$

(iii) the lower barrier L, a $\mathcal{F}_t$-adapted continuous process, s.t. $E(\sup_{0\le t\le T}(L_t)^2) < +\infty$. The solution of RBSDE w.r.t. the lower barrier L is the triple $(Y, Z, A) \in S^2[0,T] \times H^2[0,T] \times S_i^2[0,T]$, which satisfies

$$Y_t = \xi + \int_t^T g(r, Y_r, Z_r)dr + A_T - A_t - \int_t^T Z_r dB_r.$$

and $L_t \le Y_t$ for $0 \le t \le T$, $\int_0^T (Y_t - L_t)dA_t = 0$, a.s.

In the paper [3] of N. El Karoui et al, the existence and uniqueness for the solution of such RBSDE is proved. Moreover, in the Section 6, they consider a very clever method, penalization method to prove the existence. This method also stir out the study for the numerical solution for RBSDE. How to calculate the numerical solution for RBSDE is written in the paper "Convergence of solutions of discrete reflected Backward SDE's" by J.Memin and S.Peng, the simulation is done by M.Xu [4].

2.2 User's guide: Reflected BSDEs

The programs are realized by Matlab's *.p files. To run these programs, Matlab 5.3 or higher version is required. The present pagage is compressed as Xurebsde.zip. After download the compressed file, you should:

1. uncompress the file Xurebsde.zip in the document C: \matlab\work (or D:\..., your Matlab is installed in the hard disk D:\).

2. Run the Matlab command window.

3. Then within this window single click "File" in the menu buttons and then, among the prompted file buttons, single click "Set Path" button. Then within the prompted "Path Browse" window, browse and add the direction C:\matlab\work\rebsde in the Path.

4. After these preparation, you can run our program: in Matlab's command window, type "rebsde" followed with a "return". Then the reflected BSDE window (figure No.2) is prompted.

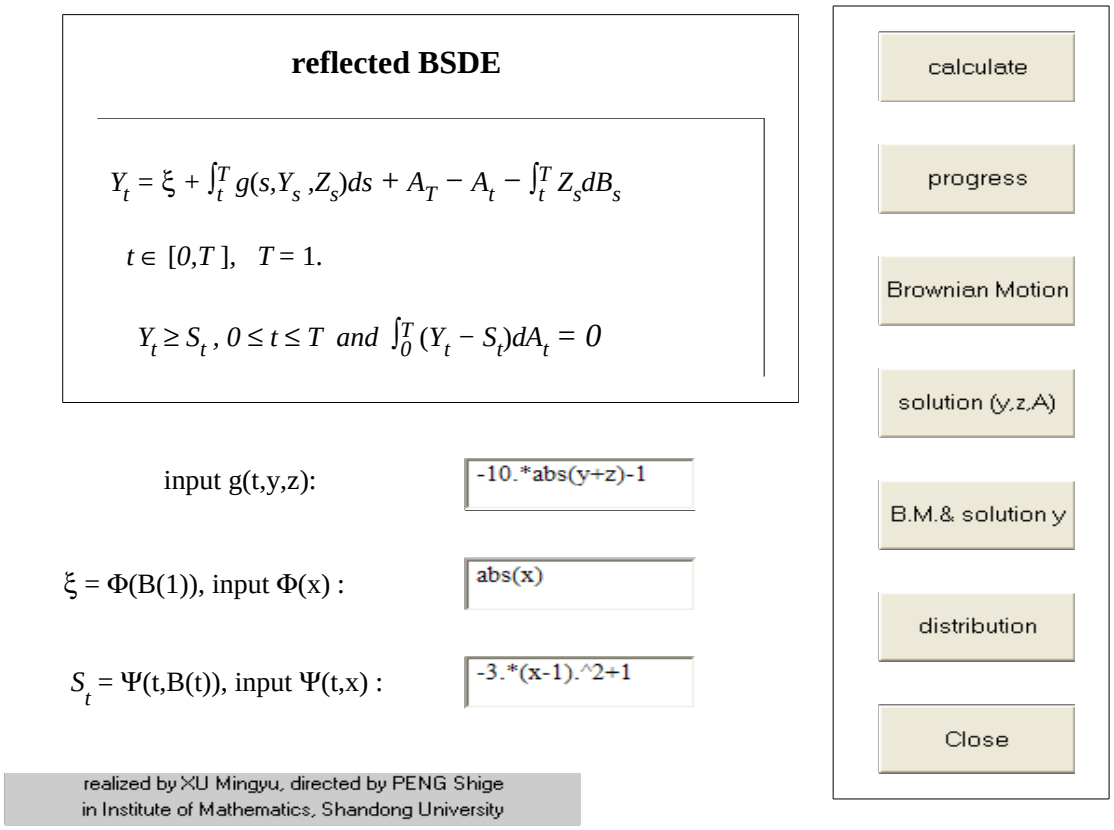
reflected BSDE
$Y_t = \xi + \int_t^T g(s,Y_s,Z_s)ds + A_T - A_t - \int_t^T Z_s dB_s$
$t \in [0,T]$, $T = 1$.
$Y_t \geq S_t$, $0 \leq t \leq T$ and $\int_0^T (Y_t - S_t)dA_t = 0$
input g(t,y,z):
-10.*abs(y+z)-1
ξ = Φ(B(1)), input Φ(x) :
abs(x)
S_t = Ψ(t,B(t)), input Ψ(t,x) :
-3.*(x-1).^2+1
realized by XU Mingyu, directed by PENG Shige
in Institute of Mathematics, Shandong University
calculate
progress
Brownian Motion
solution (y,z,A)
B.M.& solution y
distribution
Close

Figure 2:Interface reflected BSDE

An important feature of this program is its strong capacity of user-interface. Run command in the command window. We get the main user-interface shown in fig. 2.

At the left side of the user-interface, the reflected BSDE is shown on the upside to indicate you the meaning of input functions. On the downside, you can input three functions: the generator g, the terminal condition ξ, and the barrier S_t. For example, in fig.1.2 we will input: $g(t,y,z) = -10\,|y+z| - 1$, $\xi = \Phi(B_1) = |B_1|$, $S_t = \Psi(t,B_t) = -3 \times (B(t)-1)^2 + 1$. So in the blank spaces we type g=-10.*abs(y+z)-1, Φ=abs(x), Ψ(t,x)=-3.*(x-1).^2+1. To see the expression in Matlab, please read the tablet in Section 2.2, or use the help in Matlab.

For the coefficient g, the function only depend on t, y, z. It can not support other variable expect t, y, z. And for the terminal condition Φ, it's same, Φ only can have one variable x, which take place of the B_1. For the barrier Ψ,

it only has the variable t, x, t is for the time, x is for the Brownian motion B_t.

After inputting the parameters, you can use these programs to do the calculation.

1. Clicking the button 'calculate' on the left-side once, then the program of calculation will run. For getting the result, calculation will take certain seconds and then indicate the end of the calculation by jumping a dialogue-box "the calculation is complete." .

2. Click the next button "progress", the program will generate a new figure, named "calculating process of solution y". On this figure, you will see the backward computation procedure of the function $y(t, x)$, dynamically and backwardly. In the figure the red lines above (resp. blue lines below) show the solution y (resp. Brownian motion). The grid surface is the barrier. At the end, we can see a vertical red line, which simify the value of solution y at time 0. Then the above colorful surface of solution y is generated in a new figure, which is named "surface for solution y". The green lines in this figure show the relation between the solution y and the Brownian motion, while the blue line below indicate the range of descret Brownian motion. By clicking the button "right", "up", you can see the surface in different direction.

The next three buttons on the main user-interface are for the simulations of the solved (y_t, z_t) .

3. Click the button "Brownian motion", a dynamically generated Brownian path will appear on the new figure, 'Sample way of Brownian Motion'. This path will terminated by a jump of a vertical line indicating the terminal value $y_T = \xi(\omega)$ of this sample. If you click the button "more" on this new figure, then another Brownian path and the related $\xi(\omega)$ will be produced in a different color.

4. Click the button "solution (y, z)" , you will see a moving (y_t, z_t, A_t) on the generated new figure "trajetories of solution y and z and A ". In the 1st (resp. the 2nd) column we show the trajetory of (B_t, y_t) (resp. (B_t, z_t)). On the above they are showed by a 3-d moving image, the red (resp. blue) lines show a trajetory of the solution y (resp. Brownian motion), and the light red vertical lines indicate the relation between the two trajetories. On the below, this trajetory of solution y is showed in 2-d moving image by a red line, with time being the x-coordinate. In the above of the 3rd column, the 'push' A_t corresponding to the solution y_t is shown. And in the below of the 3rd column, it's the difference of the y_t and S_t, i.e. $y_t - S_t$. Clicking "more" button on this figure, there will produce a different triple (y_t, z_t, A_t)

corresponding to a different Brownian motion path. Like in the figure for the solution surface the button "center", "right" and "up" are for see the two 3-d image in different direction.

5. Clicking the following button "B.M. and solution y", you will get a new figure, "solution y on the surface", for the sample way of solution y on the solution surface. On the above window of it, a trajectory of Brownian Motion $B_t(\omega)$ is showed on the ground, while the solution y_t according to this Brownian motion is showed on the solution surface. And the grey surface is for the barrier S_t. On the below windows, there are the trajectories of A_t (on the left) and $y_t - S_t$ (on the right). The button "more" is still for a new group of lines produced in the different color, and the button "right", "up" are for the same use like the ones in the figure "surface for solution y".

6. Click the next button "distribution", a new figure "distribution function" is generated to show the distribution function of solution y at different time t. On this figure, from the time $T = 1$ to $t = 0$, the distribution functions of solution y are showed by lines in different color by turn. The button "right", "up" are for the same use.

7. For closing the figures there is two ways. One is using the button "Close" on them. The other is to click the little cross on the right-up corner of the figure.

In all the images,the meanings of the coordinate axises are noted directly beside the corresponding axises. So we omit the detailed explanation.

2.3 User's guide: the penalization BSDEs for RBSDEs

For an RBSDE with the parameters (ξ, g, L) satisfying the assumptions (i)-(iii) in section 2.1, denote $(Y^p, Z^p) \in S^2[0,T] \times H^2[0,T]$ (for each $p \in \mathbf{N}$) be the unique pair of $\mathcal{F}_t$-progressively processes satisfying

$$Y_t^p = \xi + \int_t^T g(r, Y_r^p, Z_r^p)dr + p\int_t^T (Y_r^p - S_r)^- dr - \int_t^T Z_r^p dB_r,$$

and let $A_t^p = \int_0^T (Y_r^p - S_r)^- dr$. Such equation is called the penalization BSDEs w.r.t. (ξ, g, L).

In [3] N.El Karoui et al. Section 6, we know that the solution (Y^p, Z^p, A^p) of the penalization BSDEs converges to the solution (Y, Z, A) of RBSDE in the space $S^2[0,T] \times H^2[0,T] \times S_i^2[0,T]$. Then we use the numerical method for BSDEs to calculate the penalization BSDEs.

The programs are realized by Matlab's *.p files. To run these programs, Matlab 5.3 or higher version is required. The present pagage is compressed as Xupebsde.zip. After download the compressed file, you should:

1. uncompress the file Xupebsde.zip in the document C: \matlab\work (or D:\..., your Matlab is installed in the hard disk D:\).

2. Run the Matlab command window.

3. Then within this window single click "File" in the menu buttons and then, among the prompted file buttons, single click "Set Path" button. Then within the prompted "Path Browse" window, browse and add the direction C:\matlab\work\pebsde in the Path.

4. After these preparation, you can run our program: in Matlab's command window, type "pebsde" followed with a "return". Then the penalization BSDE window (figure No.3) is prompted.

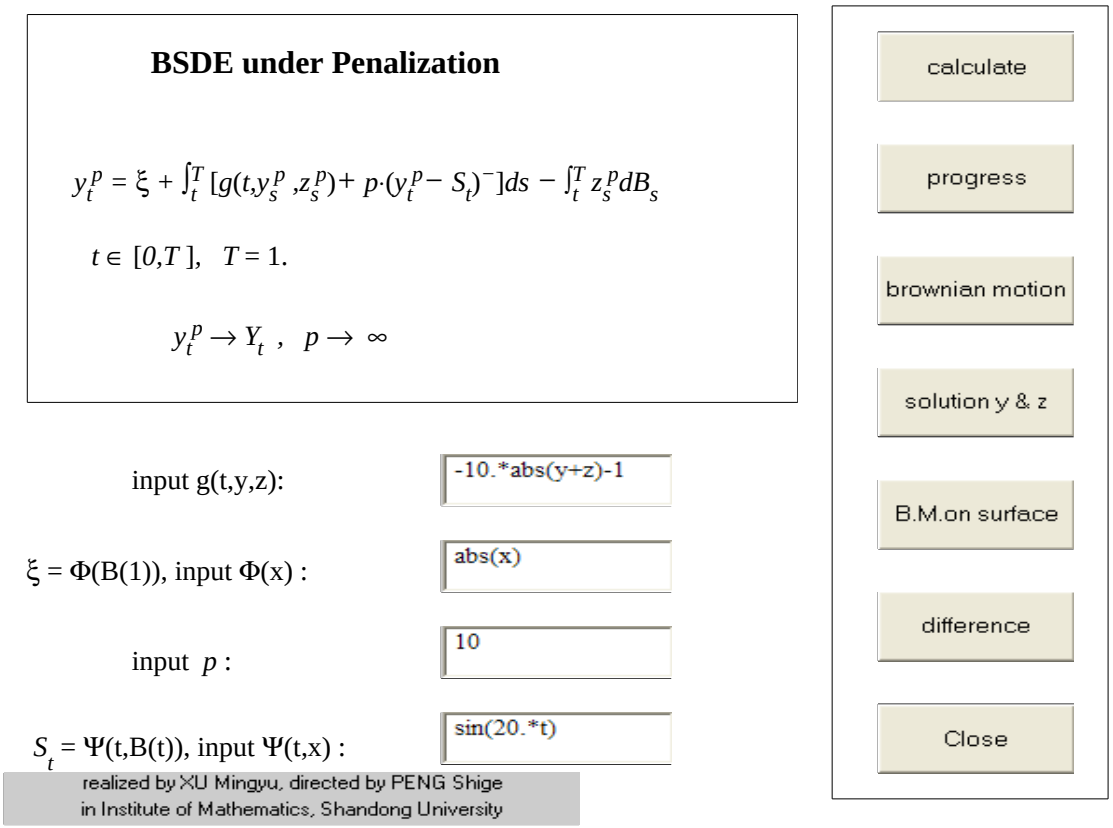

Figure 3:Interface penalization BSDE

At the left side of the user-interface, the penalization BSDE is shown on the upside to indicate you the meaning of input functions. On the downside, you can input three functions: the generator g, the penalization's parameter p, the terminal condition ξ, and the barrier S_t. For example, in fig.1.2 we will input: $g(t,y,z) = -10\,|y+z| - 1$, $\xi = \Phi(B_1) = |B_1|$, $p = 10$, $S_t = \Psi(t, B_t) = -3 \times (B(t)-1)^2 + 1$. So in the blank spaces we type g=-10.*abs(y+z)-1, Φ=abs(x), p=10, Ψ(t,x)=-3.*(x-1).^2+1. To see the expression in Matlab, please read the tablet in Section 2.2, or use the help in Matlab.

For the coefficient g, the function only depend on t, y, z. It can not support other variable expect t, y, z. And for the terminal condition Φ, it's same, Φ only can have one variable x, which take place of the B_1. For the barrier Ψ, it only has the variable t, x, t is for the time, x is for the Brownian motion B_t.

After inputting the parameters, you can use these programs to do the calculation. On this figure the functions of the buttons on the right side are almost same with the ones on the figure for 'RBSDE'. You can find the detail explanation in the Section 2.2.

1. The button 'calculate' is for calculation.

2. The button "progress" is for show the procedure of the function $y^p(t,x)$, dynamically and backwardly.

The next three buttons on the main user-interface are for the simulations of the solved (y_t^p, z_t^p) .

3. The button "Brownian motion" is for a dynamically generated Brownian with the terminal value $y_T^p = \xi(\omega)$ of this sample.

4. The button "solution (y,z)" , is for trajetories of solution y^p and z^p. Same as the one on the figure "g-expectation".

5. The button "B.M. and solution y" is for the sample way of solution y^p on the solution surface. The differences with the button "B.M. and solution y" is the window below, this window is for $A_t^p = \int_0^T (Y_r^p - S_r)^- dr$.

6. The next button "difference" is to show the difference between the numerical solution of penelization BSDE and reflected BSDE. After load the data of penalization BSDE and reflected BSDE, first, the program will check if the parametres g, ξ, S, of the two equations are same, if they are not same a dialogue window will be generated "CANNOT COMPARE! The function or terminal condition or obstacle is defferent!". If they are same, a new figure named "the difference of PBSDE and RBSDE" will be shown, with the surface of the difference. The button "right", "up" are for the same use.

7. For closing the figures there is two ways. One is using the button

"Close" on them. The other is to click the little cross on the right-up corner of the figure.

In all the images,the meanings of the coordinate axises are noted directly beside the corresponding axises. So we omit the detailed explanation.

References

[1] P.Briand, B. Delyon, J. Mémin. *Donsker-type theorem for BSDEs,* Elect. Comm. in Probab. 6 (2001) 1-14.

[2] J. M. Bismut (1973): "Conjugate Convex Functions in Optimal Stochastic Control," *J.Math. Anal. Apl.*, **44**, pp.384-404.27.

[3] N. El Karoui, C. Kapoudjian, E. Pardoux, S. Peng and M.-C. Quenez (1997), *Reflected Solutions of Backward SDE and Related Obstacle Problems for PDEs*, Ann. Probab. **25**, no 2, 702–737.

[4] J.Mémin, S.Peng, (simulatiob by M.Xu) (2002) *Convergence of solutions of discret reflected backward SDE's,* preprint.

[5] E. Pardoux and S. Peng. *Adapted solutions of Backward Stochastic Differential Equations.* Systems Control Lett. 14 (1990), 51-61.